



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/779,855	02/17/2004	Gosagan Padmanabhan	852663.405	6239

38106 7590 06/11/2007  
SEED INTELLECTUAL PROPERTY LAW GROUP PLLC  
701 FIFTH AVENUE, SUITE 5400  
SEATTLE, WA 98104-7092

EXAMINER

FENNEMA, ROBERT E

ART UNIT	PAPER NUMBER
----------	--------------

2183

MAIL DATE	DELIVERY MODE
-----------	---------------

06/11/2007

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/779,855	<b>Applicant(s)</b> PADMANABHAN ET AL.	
	<b>Examiner</b> Robert E. Fennema	<b>Art Unit</b> 2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) ☒ Responsive to communication(s) filed on 3/27/2007.
- 2a) ☒ This action is **FINAL**.                      2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) ☒ Claim(s) 1-24 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-24 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)   | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                       | 5) <input type="checkbox"/> Notice of Informal Patent Application                       |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

**DETAILED ACTION**

1. Claims 1-24 have been considered. Claims 20-24 added as per Applicant's request. Claim 6 amended as per Applicant's request.

***Claim Rejections - 35 USC § 103***

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 1-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Applicants admitted prior art (herein Padmanabhan), in view of Christie (International Publication WO 02/13005), further in view of Pilat et al. (USPN 4,448,173)

4. As per Claim 1, Padmanabhan teaches: A microprocessor (Page 1, Line 12) comprising:

a memory array having a stack for saving contextual data (Page 1, Lines 16-17);

a central processing unit coupled to the memory array (Page 1, Line 13), the central processing unit having registers containing contextual data (Page 1, Lines 22-23) and a stack pointer (Page 2 Lines 19-23) and being arranged for saving contextual data upon a switch from a first to a second program (Page 2, Lines 10-23), but fails to teach:

in a variable number of registers that varies according to the value of at least one flag stored in a register to be saved.

Christie teaches a computer system which implements an extended register set, allowing for the use of additional registers, allowing more operands to be stored in fast memory, as opposed to main memory, which is much slower (Page 2, Lines 15-19). A control register holds flags which determines if the current process is using the extended mode registers or not, a register which is saved in a context switch (Page 3, Lines 15-19). However, Christie teaches saving every single register in a context switch (Page 11, Lines 14-32), therefore, while Christie teaches an advantageous method to increase performance and allow for more use of fast register memory, Christie does not teach that this flag indicating the use of extended register memory can be used to vary the number of registers saved in a context switch.

However, Pilat teaches a system in which variable amounts of state exist, and teaches that it would be wasteful to store excess data for operations which do not require said data (Column 3, Lines 13-23, and Column 4, Lines 20-33). In addition, as extrinsic evidence, Examiner refers to Shaylor et al. (USPN 6,408,325), which teaches the use of dirty bits, and on a context switch, only saving those registers which are dirty (have been modified), as saving all registers creates a large memory overhead which is undesirable, and it additionally causes a delay in allowing a new context to resume, thus, saving unnecessary data is taught as an extreme hindrance to processor operation (Column 2, Lines 27-54). Pilat solves this issue by having two different call instructions, which indicate if only basic, or the basic and extended versions of data are

Art Unit: 2183

required to be saved (Column 5, Lines 19-27). In the basic case, only a few values are saved, in the general case, the extended versions are saved. Given these teachings of saving only data which is required to be saved, and specifically, saving extended data only if it is in use, one of ordinary skill in the art would have recognized that the control register's flag indicating if the extended mode registers were in use could be used as an indication of whether or not the extended registers needed to be saved on a context switch, as disabled registers would not have a state required to be stored. Given the advantages of saving as little data as is required, one of ordinary skill in the art at the time the invention was made would have been motivated to combine Pilat's teachings of storing either regular data, or regular data in addition to extended data, with Christies teachings of an extended register set with an extended mode enable flag, to not save extended register data when the extended registers were not enabled, avoiding the memory congestion and processor delays caused by saving all registers as is taught in Christie and Padmanabhan alone.

5. As per Claim 2, Christie teaches: The microprocessor according to claim 1 wherein the central processing unit is arranged for changing the value of the flag according to the content of a register, before saving contextual data contained in a variable number of registers that varies according to the value of the flag (Page 11, Lines 14-16, in a context switch, the processor saves the appropriate registers, all flags would be set before saving).

6. As per Claim 3, Christie teaches: The microprocessor according to claim 2 wherein the central processing unit is arranged for changing the value of the flag according of the content of an extended addressing register of a program counter of the central processing unit.(Page 3, Lines 15-19, if one of the extended addressing registers have content (valid data), then they've been enabled).

7. As per Claim 4, Christie teaches: The microprocessor according to claim 3 wherein the central processing unit is arranged for:

when the content of the extended addressing register is equal to 0, saving all the registers of the central processing unit containing contextual data, except for the extended addressing register,

when the content of the extended addressing register is not equal to 0, saving all the registers of the central processing unit containing contextual data, including the extended addressing register.

As explained in the rejections for Claims 1-3, if the extended addressing registers have data, then they have been enabled as taught by Christie. Given the combination with the other references, in which the extended registers are backed up only if they are enabled, then in the case when the content is not equal to zero, saving all of the registers including the extended addressing register would occur as explained above. While the specific teachings of what happens when the content is equal to 0 is not taught by the references (saving all but a single register), it is not required that the value would ever be zero, it is very possible that the value would always be non-zero, thus the

Art Unit: 2183

references read on the claim in the situation that the content of the extended addressing register is always the same, and in this particular example, non-zero.

8. As per Claim 5, Pilat teaches: The microprocessor according to claim 1 wherein the central processing unit is arranged for performing a test on the value of the flag so as to determine the number of registers to be saved (Column 5, Lines 5-13, the mode of operation must be tested to determine how much data to save).

9. As per Claim 6, Pilat teaches: The microprocessor according to claim 1 wherein the central processing unit is arranged for, upon the return to the first program:

restoring the register containing the flag at a first time (Column 4, Lines 38-39, a return must return all previously saved states); and

restoring contextual data contained in a variable number of registers that varies according to the value of the flag present in the restored register at a second time subsequent to the first time (Column 5, Lines 29-34).

10. As per Claim 7, Christie teaches: The microprocessor according to claim 1 wherein the central processing unit is arranged for saving the register containing the flag last (Page 11, Lines 14-20).

Art Unit: 2183

11. As per Claim 8, Christie teaches: The microprocessor according to claim 1 wherein the flag comprises at least one bit of a register containing condition code flags (Page 3, Lines 15-19).

12. As per Claim 9, Padmanabhan teaches: A method for managing the stack of a microprocessor having a central processing unit (Page 1, Line 13) and a memory array (Page 1, Lines 16-17), the central processing unit having registers containing contextual data (Page 1, Lines 22-23) and a stack pointer (Page 2, Lines 19-23), the stack being a zone of the memory array dedicated to saving contextual data upon a switch from a first to a second program (Page 2, Lines 10-23), but fails to teach:

saving contextual data contained in a variable number of registers that varies according to the value of at least one flag stored in a register to be saved.

Christie teaches a computer system which implements an extended register set, allowing for the use of additional registers, allowing more operands to be stored in fast memory, as opposed to main memory, which is much slower (Page 2, Lines 15-19). A control register holds flags which determines if the current process is using the extended mode registers or not, a register which is saved in a context switch (Page 3, Lines 15-19). However, Christie teaches saving every single register in a context switch (Page 11, Lines 14-32), therefore, while Christie teaches an advantageous method to increase performance and allow for more use of fast register memory, Christie does not teach that this flag indicating the use of extended register memory can be used to vary the number of registers saved in a context switch.



However, Pilat teaches a system in which variable amounts of state exist, and teaches that it would be wasteful to store excess data for operations which do not require said data (Column 3, Lines 13-23, and Column 4, Lines 20-33). In addition, as extrinsic evidence, Examiner refers to Shaylor et al. (USPN 6,408,325), which teaches the use of dirty bits, and on a context switch, only saving those registers which are dirty (have been modified), as saving all registers creates a large memory overhead which is undesirable, and it additionally causes a delay in allowing a new context to resume, thus, saving unnecessary data is taught as an extreme hindrance to processor operation (Column 2, Lines 27-54). Pilat solves this issue by having two different call instructions, which indicate if only basic, or the basic and extended versions of data are required to be saved (Column 5, Lines 19-27). In the basic case, only a few values are saved, in the general case, the extended versions are saved. Given these teachings of saving only data which is required to be saved, and specifically, saving extended data only if it is in use, one of ordinary skill in the art would have recognized that the control register's flag indicating if the extended mode registers were in use could be used as an indication of whether or not the extended registers needed to be saved on a context switch, as disabled registers would not have a state required to be stored. Given the advantages of saving as little data as is required, one of ordinary skill in the art at the time the invention was made would have been motivated to combine Pilat's teachings of storing either regular data, or regular data in addition to extended data, with Christies teachings of an extended register set with an extended mode enable flag, to not save extended register data when the extended registers were not enabled, avoiding the

Art Unit: 2183

memory congestion and processor delays caused by saving all registers as is taught in Christie alone.

13. As per Claim 10, Christie teaches: The method according to claim 9, comprising a step of:

changing the value of the flag according to the content of a register, before saving contextual data contained in a variable number of registers that varies according to the value of the flag (Page 11, Lines 14-16, in a context switch, the processor saves the appropriate registers, all flags would be set before saving).

As per Claim 11, Christie teaches: the method according to claim 10 wherein the value of the flag is changed according to the content of an extended addressing register of a program counter of the central processing unit (Page 3, Lines 15-19, if one of the extended addressing registers have content (valid data), then they've been enabled).

14. As per Claim 12, Christie teaches: The method according to claim 11, comprising the following steps:

when the content of the extended addressing register is equal to 0, saving all the registers of the central processing unit containing contextual data, except for the extended addressing register,

when the content of the extended addressing register is not equal to 0, saving all the registers of the central processing unit containing contextual data, including the extended addressing register.

As explained in the rejections for Claims 9-11, if the extended addressing registers have data, then they have been enabled as taught by Christie. Given the combination with the other references, in which the extended registers are backed up only if they are enabled, then in the case when the content is not equal to zero, saving all of the registers including the extended addressing register would occur as explained above. While the specific teachings of what happens when the content is equal to 0 is not taught by the references (saving all but a single register), it is not required that the value would ever be zero, it is very possible that the value would always be non-zero, thus the references read on the claim in the situation that the content of the extended addressing register is always the same, and in this particular example, non-zero.

15. As per Claim 13, Pilat teaches: The method according to claim 9, comprising a step of:

testing the value of the flag for determining the number of registers containing the data to be saved (Column 5, Lines 5-13, the mode of operation must be tested to determine how much data to save).

16. As per Claim 14, Pilat teaches: The method according to claim 9, comprising the following steps:

Art Unit: 2183

restoring the register containing the flag (Column 4, Lines 38-39, a return must return all previously saved states); then

restoring contextual data contained in a variable number of registers that varies according to the value of the flag present in the restored register (Column 5, Lines 29-34).

17. As per Claim 15, Christie teaches: The method according to one claim 9 wherein the register containing the flag is saved last and is restored first (Page 11, Lines 14-20).

18. As per Claim 16, Christie teaches: The method according to claim 9 wherein the flag is formed by at least one bit of a register containing condition code flags (Page 3, Lines 15-19).

19. As per Claim 17, Padmanabhan teaches: A microprocessor comprising:  
a memory array having stored therein contextual data (Page 1, Lines 16-17);  
a central processing unit coupled to the memory array (Page 1 Line 13);  
a plurality of registers associated with the central processing unit (Page 1, Lines 22-23); and

a stack pointer associated with the central processing unit and being arranged for directing contextual data to be stored (Page 2, Lines 19-23), but fails to teach:

a first group of the registers storing contextual data and a second group of the registers not storing contextual data when a flag has a first value and switching to store

contextual data also in the second group of registers when the flag switches to a second value, such that the number of registers that store contextual data is variable;

and where the flag is stored in a register to be saved as part of the program contextual data;

a stack pointer associated with the central processing unit and being arranged for directing contextual data to be stored in the first group only or in both the second group and the first group, based on the flag value.

Christie teaches a computer system which implements an extended register set, allowing for the use of additional registers, allowing more operands to be stored in fast memory, as opposed to main memory, which is much slower (Page 2, Lines 15-19). A control register holds flags which determines if the current process is using the extended mode registers or not, a register which is saved in a context switch (Page 3, Lines 15-19). However, Christie teaches saving every single register in a context switch (Page 11, Lines 14-32), therefore, while Christie teaches an advantageous method to increase performance and allow for more use of fast register memory, Christie does not teach that this flag indicating the use of extended register memory can be used to vary the number of registers saved in a context switch.

However, Pilat teaches a system in which variable amounts of state exist, and teaches that it would be wasteful to store excess data for operations which do not require said data (Column 3, Lines 13-23, and Column 4, Lines 20-33). In addition, as extrinsic evidence, Examiner refers to Shaylor et al. (USPN 6,408,325), which teaches the use of dirty bits, and on a context switch, only saving those registers which are dirty

Art Unit: 2183

(have been modified), as saving all registers creates a large memory overhead which is undesirable, and it additionally causes a delay in allowing a new context to resume, thus, saving unnecessary data is taught as an extreme hindrance to processor operation (Column 2, Lines 27-54). Pilat solves this issue by having two different call instructions, which indicate if only basic, or the basic and extended versions of data are required to be saved (Column 5, Lines 19-27). In the basic case, only a few values are saved, in the general case, the extended versions are saved. Given these teachings of saving only data which is required to be saved, and specifically, saving extended data only if it is in use, one of ordinary skill in the art would have recognized that the control register's flag indicating if the extended mode registers were in use could be used as an indication of whether or not the extended registers needed to be saved on a context switch, as disabled registers would not have a state required to be stored. Given the advantages of saving as little data as is required, one of ordinary skill in the art at the time the invention was made would have been motivated to combine Pilat's teachings of storing either regular data, or regular data in addition to extended data, with Christies teachings of an extended register set with an extended mode enable flag, to not save extended register data when the extended registers were not enabled, avoiding the memory congestion and processor delays caused by saving all registers as is taught in Christie alone.

Art Unit: 2183

20. As per Claim 18, Christie teaches the microprocessor according to claim 17 wherein the second group of registers includes a register which is used as an extended addressing register when the flag is at a first value (Page 3, Lines 15-19).

21. As per Claim 19, Christie teaches: The microprocessor according to claim 17 wherein the second group of registers includes a single register (Page 3, Lines 6-7).

22. As per Claim 20, Padmanabhan teaches: A microprocessor comprising:  
a memory array (Page 1, Lines 16-17);  
a central processing unit coupled to the memory array (Page 1, Line 13);  
a first group of registers associated with the central processing unit and arranged to store contextual data (Page 1, Lines 22-23), but fails to teach:

a second group of registers associated with the central processing unit and arranged to store contextual or non-contextual data; and

a flag defined in the first group of registers;

wherein the central processing unit is arranged to suspend execution of a first program and commence execution of a second program and the central processing unit is further arranged to store the first group of registers in the memory array and conditionally store a variable number of registers of the second group of registers in the memory array.

Christie teaches a computer system which implements an extended register set, allowing for the use of additional registers , allowing more operands to be

Art Unit: 2183

stored in fast memory, as opposed to main memory, which is much slower (Page 2, Lines 15-19). A control register holds flags which determines if the current process is using the extended mode registers or not, a register which is saved in a context switch (Page 3, Lines 15-19). However, Christie teaches saving every single register in a context switch (Page 11, Lines 14-32), therefore, while Christie teaches an advantageous method to increase performance and allow for more use of fast register memory, Christie does not teach that this flag indicating the use of extended register memory can be used to vary the number of registers saved in a context switch.

However, Pilat teaches a system in which variable amounts of state exist, and teaches that it would be wasteful to store excess data for operations which do not require said data (Column 3, Lines 13-23, and Column 4, Lines 20-33). In addition, as extrinsic evidence, Examiner refers to Shaylor et al. (USPN 6,408,325), which teaches the use of dirty bits, and on a context switch, only saving those registers which are dirty (have been modified), as saving all registers creates a large memory overhead which is undesirable, and it additionally causes a delay in allowing a new context to resume, thus, saving unnecessary data is taught as an extreme hindrance to processor operation (Column 2, Lines 27-54). Pilat solves this issue by having two different call instructions, which indicate if only basic, or the basic and extended versions of data are required to be saved (Column 5, Lines 19-27). In the basic case, only a few values are saved, in the general case, the extended versions are saved. Given these teachings of saving only data which is required to be saved, and specifically, saving extended data only if it is in use, one of ordinary skill in the art would have recognized that the control



Art Unit: 2183

register's flag indicating if the extended mode registers were in use could be used as an indication of whether or not the extended registers needed to be saved on a context switch, as disabled registers would not have a state required to be stored. Given the advantages of saving as little data as is required, one of ordinary skill in the art at the time the invention was made would have been motivated to combine Pilat's teachings of storing either regular data, or regular data in addition to extended data, with Christies teachings of an extended register set with an extended mode enable flag, to not save extended register data when the extended registers were not enabled, avoiding the memory congestion and processor delays caused by saving all registers as is taught in Christie alone.

23. As per Claim 21, Christie teaches: The microprocessor according to claim 20 wherein the flag is used to determine if the second group of registers are arranged to store contextual data (Page 3, Lines 15-16, if the extended register mode is not enabled, the extended registers contain garbage (non-contextual) information, and vice versa).

24. As per Claim 22, Pilat teaches: The microprocessor according to claim 21 wherein the second group of registers is only stored in the memory array if the flag has a first value (In Column 5, Lines 5-13, Pilat shows that only registers that need to be saved (ones in use) for performance purposes, and as outlined in the independent

claim, the combination of Christie and Pilat would result in only saving the extended registers if they were in fact in use).

25. As per Claim 23, Christie teaches: The microprocessor according to claim 20 wherein the flag is used to indicated if the second group of registers are arranged to store contextual data (Page 3, Lines 15-16, if the extended registers are not in use, they cannot contain contextual data).

26. As per Claim 24, Christie teaches: The microprocessor according to claim 23 wherein the flag is loaded with a first value only if the second group of registers are arranged to store contextual data (Page 3, Lines 15-16, the only way the second group of registers can contain contextual data is if the flag is set to indicate them in use).

### ***Response to Arguments***

27. Applicant's arguments filed 3/27/2007 have been fully considered but they are not persuasive.

28. Regarding Applicants remarks as to independent claims 1 and 9, Applicant and Examiner are in agreement about section A. In section B, Applicant has argued that since Christie has taught that 32 bit operating systems may not be able to recognize extended registers, that it would not be aware of the flag bit indicating the operating mode. Examiner agrees that Christie does not teach saving a variable number of

Art Unit: 2183

registers, which is why Pilat is used in the rejection as well, Christie is used to primarily teach the advantage of using extended registers. However, Examiner notes that when Christie is operating in non-32 bit modes, the issue the Applicant has brought up cannot occur, and in that case the argument is moot. Regarding Section C (and D, as they deal with the same issue), Applicant has argued that the registers containing the indication of operating mode are not saved, and has argued that the register file is saved, and that the registers containing the flags are not, because they are in the decode unit. While Examiner concedes that Christie does not explicitly state that the Control and Flags registers are included, Christie does state on Page 11 that: "a typical 32-bit operation system saves state information of CPU 32 needed to restore the interrupted program or task". As Christie has stated, on a context switch, it is not just the register file that is backed up, but all registers containing state information, which is also extremely well known in the art. As indicated on Page 3, the flags register stores information indicating whether or not the extended register mode is enabled by a current process. Examiner asserts that this is critically important state information, and must have been saved, because if this data is not saved on a context switch, upon restore, the extended registers may not be able to be accessed by the process that requires them (additionally, in the combination that Examiner has laid out in the rejection, these values become even more crucial, and thus making it even more important that they be saved). Therefore, Examiner disagrees with Applicants assertion that these crucial registers containing necessary state information for the process are not saved.

Art Unit: 2183

29. Now referring to Section E, Applicant has argued that Pilat sets his flag different than the present claims, which should result in the rejection being removed. However, Examiner feels that Applicant is attempting to bodily incorporate the entire reference of Pilat into the rejection, which is improper. Pilat was used to teach why it would be advantageous to not store registers which have not been modified, because the speed of context switches can be drastically increased by doing so, and the combination of Christie and Pilat showed how the flag would store variable numbers of registers. Referring to Section F, Applicant has argued that Shaylor is directed to a different field than the claims. Examiner used Shaylor to teach dirty bits, an extremely well known concept in the art, for both VLIW and non-VLIW processors, and was used to help illustrate this well known concept to the Applicants for their benefit. Again, the fact that Shaylor is a VLIW machine has nothing at all to do with the rejection, Examiner only referenced Shaylor to help explain this concept to the Applicants.

30. Regarding Independent Claim 17, the arguments that applied to the previous claims apply here as well, and Examiner refers Applicants to the above remarks to address the concerns with this claim.

31. Regarding Claims 2 and 10, the claim recites that the value of the flag is changed according to the content of a register, before being saved. When the extended registers are in use, the extended register contents have data, meaning the flag to enable extended mode must be set, all of which must be done before being saved.

32. Regarding Claims 3 and 11, Applicant appears to be arguing that Christie does not teach a program counter register, which is a completely unreasonable argument, as a program counter is a vital and inherent part of computer systems, and something to that effect must be present, in Christies case, it appears to be the base and index registers, EBP for example being the base pointer. Additionally, Examiner refers Applicant to the remarks for Claims 2 and 10 for the remainder of the arguments to these claims.

33. Regarding Claims 4 and 12, the way the claims have been laid out, using the language "when" as a conditional, the art only has to read on one of the limitations, because the other conditional statement (if 0, or if not 0) is not guaranteed to occur, thus given the claim language, does not have to occur, and therefore the combination of references does not have to read on a limitation which will not apply, as long as they read on at least one of the conditional limitations.

34. Regarding Claims 7 and 15, Christie does not teach away from the register containing the flag last, in fact, Christie does not teach any specific ordering, thus the ordering could be considered random. In the particular instance that the register containing the flag is saved last, Christie reads on the claim.

Art Unit: 2183

35. Regarding the new claims, Applicant again appears to be bodily incorporating the references into each other, instead of looking at the rejection and combination that the Examiner laid out, which results in the combination saving a variable number of registers. Examiner has laid out how in the combination of references a variable number of registers being saved would occur.

### ***Conclusion***

36. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Robert E. Fennema whose telephone number is (571) 272-2748. The examiner can normally be reached on Monday-Friday, 8:45-6:15.

Art Unit: 2183

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Robert E Fennema  
Examiner  
Art Unit 2183

RF



EDDIE CHAN  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100